

automatic filler 218 which fills out the form under the guidance of the rules engine 220. The rules engine 220 retrieves from the dictionary 1100 the rule specifically applicable to completing each field of this form and applies the rule using personal information that is retrieved from the wallet database 1100 entries for this particular user. The completed form is then returned to the match engine 500.

[00091] If a dictionary 1000 entry for this form is not found, then an attempt is made to fill out the form using rules in the dictionary that may not match the form precisely and that may be somewhat contradictory because they originated from many different forms. At step 610, the fuzzy logic 226, working with the labels and sizes of fields in the form, tries to find the best match between those labels and field sizes and the rule information contained within the dictionary 1000. The fuzzy logic 226 may compare the spelling of rule labels to those found in the form and find the closest match in that regard, if an exact match cannot be found. In this manner, the fuzzy logic system generates a new tentative set of rules and stores them in a temporary storage area 228. If this effort is successful, then program control continues its step 606 where the automatic filler 218 is called upon to fill out the form using the rules engine 220 governed this time by the tentative rule set in the storage area 228. Once again, the completed form is returned to the match engine 500

[00092] After the form passes through the automatic filler and if the fuzzy logic system is unable to come up with good matches between existing rules and the names and sizes of the fields in the new form, then (step 702 in Figure 7) a check is made by the history unit 230 to see if a copy of the form exists in the history database 1200. If so, information comparable to that shown at 1202 in Figure 12 is retrieved from the history database entry for that form and is used to guide the completion of the form with personal information taken from the wallet database 1100 of the user (step 704). The completed form is then returned from the match engine to the form fill proxy 400 (step 706), which returns it to the user's browser 108 or 114 for review and possible revision.

[00093] If no prior copy of the same form exists in the history database, then at step 708 in Figure 7, the form is simply not completed but is returned from the match engine to the form fill proxy 400 which sends on the incomplete form to the user's browser 108 or 114 to be filled out completely by the user.

#### THE COMPLETED FORM ANALYSIS ENGINE

[00094] Figure 8 presents a block diagram of the steps performed by the completed form analysis engine 800, which is called upon by the form fill proxy 400 to evaluate a form

after the user has reviewed and possibly revised it, to note any revisions and to correct and improve the future ability of the form fill system 200 to complete this and other forms.

[00095] Program control begins with one of the entry points 802, 804, 806, or 808 depending upon how this particular form was treated previously by the match engine 500. The entry point 802 corresponds to the situation when the form was filled out by the automatic filler 218 and rules engine 220 without the assistance of either the fuzzy logic 226 or the history unit 230 and the user made no modifications. In that case, no further processing is needed.

[00096] Entry point 804 is taken if a particular form was filled out using the fuzzy logic 226. In that case, program control advances from the completed form analysis engine 800 to a form field compare program 1300.

[00097] The entry point 806 corresponds to the case when the match engine 500 found the form within the history database 1200 and filled the form in automatically using a prior filled-out copy of the form as a guide. In this case, at step 822, the completed form is saved in the history database, as will be explained below (Figure 9).

[00098] Next, at step 824, all copies of the same form are retrieved from the history database 1200 and are compared, field by field, at step 826, by the history form compare program 233 to each other and to the newly completed form. Each field in each form is compared to the fields in the remaining forms. The history form compare program 233 asks the question: Is the same non-personal or symbolic data identifier linked to the same form field label in a majority of the forms or (preferably) in all of the forms? If not, then the forms are left in the history database 1200, and program control returns to the form fill proxy program 300.

[00099] If, for each and every field, the majority of the form field entries match, then at step 828, the new rule generator 234 generates rules governing the filling of the fields of this new form; and the generator 234 adds the new rules to the dictionary database 1000 along with the web address of the form and the list of rules. Finally, the copies of this form in the history database 1200, having served their purpose to govern the generation of new rules, are deleted from the history database 1200. Then program control terminates in the program 200 and is returned to the form fill proxy program 400.

[000100] Entry point 808 corresponds to the case where the form is returned blank and filled in entirely by the user. In this situation the form is transferred to the save in history database 900.

SAVING A FORM IN THE HISTORY DATABASE

[000101] Figure 9 presents a block diagram of the save in history database program 900 that saves a completed form that has been reviewed and analyzed by the user in the history database 1200. The program 900 begins at step 902 by scanning the form, locating all of the field labels or identifiers. It then locates the corresponding literal personal information in each field of the form and, by looking up that personal information in the wallet database 1100 (1102 in Figure 11) for this user, takes out the personal information such as "Jerry" and replaces it with non-personal identifiers or symbolic names for the information, such as '<FirstName>', and thereby transforms the actual completed form into a representation of the form such as that shown in 1202 in Figure 12, where the web address of the form is followed by its field labels each linked to the non-personal identifier or symbolic name of the content of that field.

[000102] Finally, at step 904, the history unit 230 is called upon to place all of this information into the history database 1200 for use in guiding form completion the next time the same form is encountered.

[000103] In the case of a first encounter with a form, such as the user name and password (or equivalent identifier) requesting form of Microsoft's HotMail service, that simply requests a user name and password (or equivalent), the literal data entered by the user may and probably will be new and will not be found in the wallet database 1100. In this special case, the history data base program 900 creates new non-personal identifiers or symbolic names for the literal data and actually commands the wallet data base to accept both the newly-created non-personal identifiers or symbolic names and the literal values and to save them as part of the data stored for this particular user. These newly created non-personal identifiers or symbolic names are then linked with the form, along with the field labels, when the form is added to the history data base. In this manner, user names and passwords (or their equivalent) may be gathered from users and saved in the wallet database 1100.

[000104] Such forms are especially marked as ones that contain "non-profile data." Accordingly, the system knows that whenever it encounters this form again for a new user, it must add new data fields to the user's wallet database 1100 entry to provide room for the storage of a new user ID and password pair (or their equivalent) for this user.

#### FORM FIELD COMPARE

[000105] Figures 13 represent the detailed diagrams of the work carried out in the Form Field Compare module. The form field compare 1300 compares an unaltered copy of the form as completed and stored in temporary storage at 244 with a copy of the user's